CVA6 CHERI Extension Support “CVA6-CHERI”

Project Concept Proposal

Bruno Sá, Sandro Pinto
Zero-Day-Labs

Peter Rugg, Jonathan Woodruff, Alexandre Joannou,
Robert N. M. Watson, Simon W. Moore
Capabilities Limited

OPENHW™

OpenHW Group

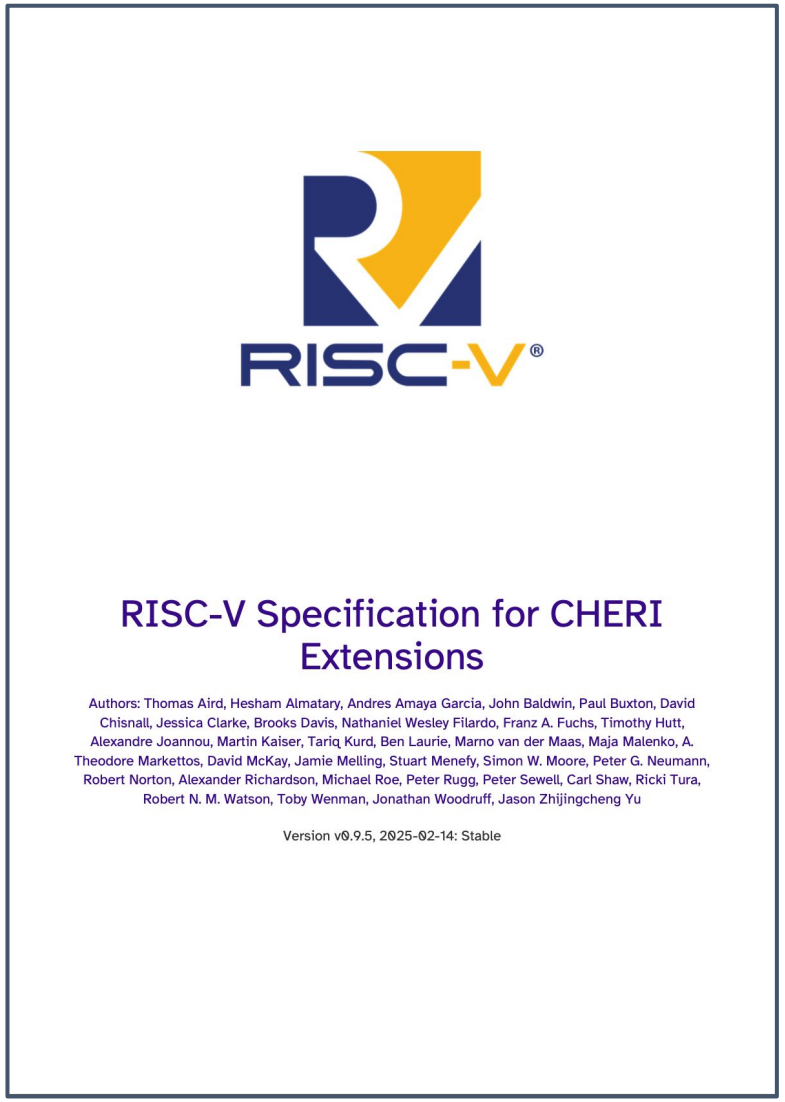
February 24th, 2025

CAPABILITIES LIMITED

Project High-Level Overview

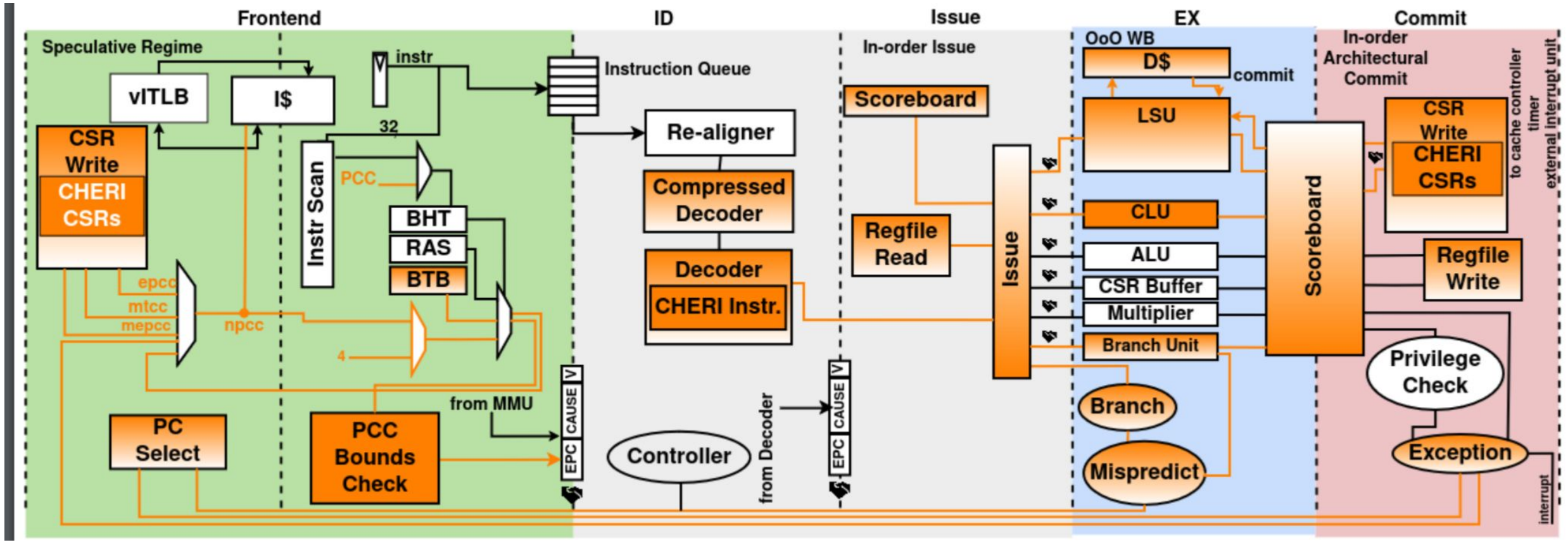
RISC-V CHERI Extension

- **The CHERI Extension entering architectural review now; could be ratified by Q4 2025**
- **Targets both embedded and application**
  - Spec decomposed into 7 extensions
  - 64-bit capabilities/32-bit address space with no MMU
  - 128-bit capabilities/64-bit address space
- **Two execution modes**
  - Capability Mode - optimized for capability pointers
  - Integer-Pointer Mode - optimized for integer pointers
- **Wider registers, new instructions:**
  - 2*XLEN-width register file
  - Capability manipulation instructions
  - 17 extended CSRs and 3 new CSRs



https://github.com/riscv/riscv-cheri/releases/download/v0.9.5/riscv-cheri.pdf

CVA6 CHERI Extension



Specification Checklist

- Zcheri Extension, Version 0.9.5
- RV64 and RV32

Feature	Status of Implementation	Target
Capability Mode	Cambridge ISAv9	Zcheri 1.0
Capability Encoding	Cambridge ISAv9 + begun conversion	Zcheri 1.0
Extended CSRs	Cambridge ISAv9	Zcheri 1.0
CHERI Instructions	Cambridge ISAv9	Zcheri 1.0
CHERI Flow Control	Cambridge ISAv9	Zcheri 1.0
Memory Bounds Checking	Yes (Cambridge ISAv9, but little change)	~Done
New Exceptions	Cambridge ISAv9	Zcheri 1.0
Page Table Entry Bits	Cambridge ISAv9	Zcheri 1.0
Tag Controller	Initial flat controller	Hierarchical + full throughput

Summary of market or input requirements

Project Requirements

- **Zcheri Extension specification version 1.0**
- **Zcheri optional via parameterization**
- **Test with TestRIG**
  - Directed-random instruction sequences with tandem verification
- **Functional validated with multiple operating systems**
  - CheriBSD
  - Cheri-linux
  - CHERI-seL4
  - CHERI-FreeRTOS
- **Test and deploy in multiple FPGAs:**
  - Genesys2
  - VCU118
- **Extend CVA6-SDK**
  - CHERI ports of OpenSBI and U-Boot

Market Requirements

- **Viable rv64 implementation**
  - Full Zcheri 1.0 specification for rv64
  - MMU with PTE.CW and PTE.UCRG support (revocation)
  - Tag Controller for DRAM with hierarchical compression
  - Timing should be equivalent to baseline
- **rv32 support is optional**
  - Market for CHERI + rv32 appears to be predominantly embedded (i.e. no MMU)
  - CHERI-Ibex/CHERIoT fills this niche
  - Would still be satisfying to allow this configuration since CVA6 includes the rv32/no MMU parameterization
  - This use case does not need a Tag Controller for DRAM, but a tagged tightly-coupled memory
- **Area optimisation is optional**
  - Many applications of CHERI-RISC-V are security processors in larger systems where performance is more important than area

Project Impact and Industry Landscape

Who would make use of OpenHW output ?

- **Chip/SoC Designers**

  Example: Google is exploring multiple avenues to adopt CHERI technology, including CHERIoT and Codasip. The chip side only wants to adopt IP for a ratified standard. Also, multiple options of compatible IP are crucial to encourage adoption.

- **Embedded System Companies**

  The Digital Security by Design program (DSbD) has produced over 25 organisations with systems ported to CHERI. Many of these would like to produce products, either based on FPGA implementations or general-purpose SoCs when they are available.

- **Software Companies**

  Companies may prove compatibility with the new Zcheri standard using CVA6-CHERI on FPGA, as it is likely to be the most accessible hardware platform that supports a full operating system.

- **Universities / Research Centers**

  CVA6-CHERI is likely to be the most convenient vehicle for further research into use of capabilities for security or performance improvement, as the code base is well known.

CHERI-RISC-V Software

Organization	Software	Status	License	Reference
University of Cambridge	FreeBSD (CheriBSD)	Complete	BSD 2-Clause	GitHub: https://github.com/CTSRD-CHERI/cheribsd
CHERI Alliance	Linux	PoC	GPLv2	GitHub: https://github.com/CHERI-Alliance/???
Capabilities Limited	seL4	PoC	BSD 2-Clause	GitHub: https://github.com/CTSRD-CHERI/seL4
Capabilities Limited	FreeRTOS	PoC	MIT	GitHub: https://github.com/CTSRD-CHERI/FreeRTOS/tree/hmka2
CHERI Alliance	QEMU	Complete	GPLv2 + BSD	GitHub: https://github.com/CHERI-Alliance/qemu
CHERI Alliance	LLVM	Complete	Apache 2.0	GitHub: https://github.com/CHERI-Alliance/llvm-project
University of Cambridge	GDB	Complete	GPLv3	GitHub: https://github.com/CTSRD-CHERI/gdb
CHERI Alliance	OpenSBI	Complete	BSD 2-Clause	GitHub: https://github.com/CHERI-Alliance/opensbi

RISC-V Cores: CHERI-RISC-V Extension

Organization	Core	Status	License	Reference
Codasip	A730	Complete	Proprietary	Website: https://codasip.com/products/application-risc-v-processors/a730
Microsoft	CHERIoT-Ibex	PoC	Apache-2.0	GitHub: https://github.com/microsoft/cheriot-ibex
University or Cambridge	CHERI-Toooba	PoC	MIT	GitHub: https://github.com/CTSRD-CHERI/Toooba
Zero-Day Labs & Capabilities Limited	CVA6	PoC	Solderpad 0.51	GitHub: https://github.com/zero-day-labs/cheri-cva6
CHERI Alliance	sail-cheri-riscv	Complete	BSD	GitHub: https://github.com/CHERI-Alliance/sail-cheri-riscv
CHERI Alliance	QEMU	Complete	GPLv2	GitHub: https://github.com/CHERI-Alliance/qemu

Why OpenHW Group should do this Project

Why OpenHW Should Do This Project ?

- **CHERI is mature**
  - The time is right for initial, commercial implementations of CHERI
  - RV64 with zCHERI has only one chip-ready implementation, Codasip A730, and it's closed source.
- **Security is a necessary tax, and companies want to share the burden**
  - Companies are being pushed for more principled security
  - CHERI is a leading technology, with extensive software support
  - Many companies want to share development effort (e.g. Microsoft, SCI, and LowRISC; CHERI Alliance)
  - OpenHW Group could be a key member of this community support
- **Risk: Zcheri ratification expected before the end of 2025, but not guaranteed**
- **Risk: We expect commercial consumers to join the effort, but not guaranteed**
  - We prefer to identify a commercial partner who intends to use and support the design before committing to upstream CHERI support to the cva6 core

Project Members, Participants, and Timeline

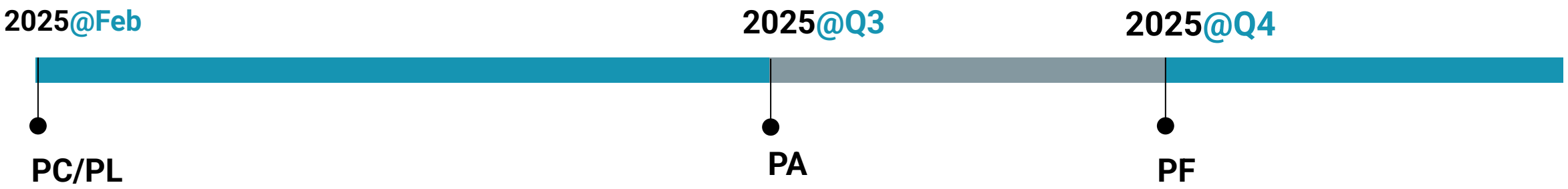
OpenHW Members Collaboration

- **Already committed:**
  - Zero-Day Labs
  - Capabilities Limited
- **Possible collaboration:**
  - Thales
  - Others Welcome!

Project Team

- **Project Leader(s):**
  - Sandro Pinto
  - Robert Watson
  - Simon Moore
- **Technical Project Leader(s):**
  - Bruno Sá
  - Jonathan Woodruff
  - Peter Rugg
  - Alexandre Joannou

Possible Project Timeline



PC: Project concept
PA: Plan approved
PL: Project launch
PF: Project freeze

Next Steps

Next Steps (3 months)

- **Integrate SystemVerilog capability library with cheri-cap-lib**
- **Update SystemVerilog capability library to Zcheri standard**
- **Update instruction decoding to Zcheri standard**
- **Implement high-performance hierarchical tag controller**
- **Bring up CheriBSD on CVA6-CHERI; both Cambridge ISAv9, and then Zcheri**

Q&A